Extracting Meanings from Simple Algorithmic Problems

Suren, Khachatryan,
American University of Armenia
Yerevan, Armenia
e-mail: skhachat@aua.am

Armen, Zakaryan
American University of Armenia
Yerevan, Armenia
armen_zakaryan@edu.aua.am

ABSTRACT

Programming contests play notable role in Computer Science education. Under the most popular format some 10 – 12 algorithmic problems are offered for coding in high-level programming languages during restricted time. Experienced participants submit the easiest problems in around 5 minutes. It supports the idea that there exist distinct patterns of recognition and classification of meanings necessary for formulation of correct solutions. These meanings are utilized as variables and algorithms. We propose a method of extraction and definition of key variables based on statistical analysis of the problem statement. At this stage we exclude natural language processing and semantic analysis. The formulated rules lead to automatic generation of variable declarations in solutions for specific class of problems. We present several case studies and discuss the strengths, weaknesses and extensions of the developed approach.

Keywords

Text processing, meaning extraction, statistics, information technology

1. INTRODUCTION

Automated meaning extraction from unstructured text has many practical applications. Three of the most popular machine learning-based methods are discussed and applied to different document types in [1]. With the current work we initiate our studies of the processing of simple programming problem statements with an ultimate goal in designing an automated solver of such problems. As the first step we focus on definitions of related variables. Extraction of definitions from regular text plays crucial role in many domains, including creation of glossaries or question answering systems [2]. We propose a method that does not look at the context of problems, but extracts candidate tokens that capture the core meanings based on descriptive statistical analysis. A context-based method of meaning extraction can be found, for example, in [3].

2. EASY ALGORITHMIC PROBLEMS

Problems offered at programming contests of the ACM ICPC style [4] consist of a preamble, the statement, specifications of the console input and output, and a sample test. When submitted to the electronic judge the time since the contest start is recorded, plus 20 minutes for each failed prior submission. The problems drastically differ in their difficulty within the same contest. We estimate the difficulty in terms of the first successful submission time in seconds.

Data from the last five NEERC contests are shown in Fig. 1. (http://neerc.ifmo.ru/information/index.html), the problems being enumerated from the most trivial to the most complicated. The time required for the first successful acceptance grows exponentially with the problem difficulty. Obviously, all stages of the software development life-cycle become time–consuming, including the meaning extraction and the requirements analysis.

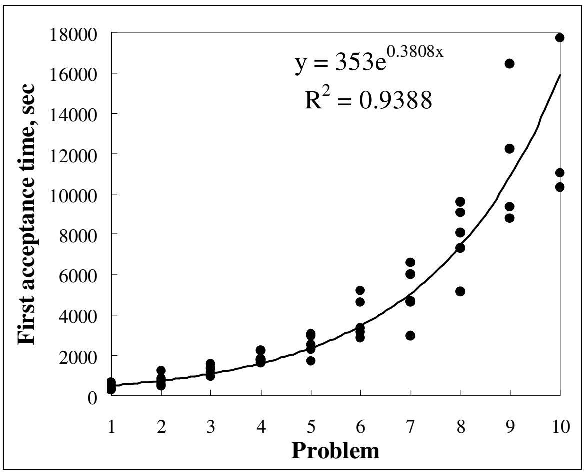

Figure 1. Time in seconds required for the first accepted submission in NEERC contest.

In the current work we focus on the simplest problems leading participants need on average around 7 minutes to solve. Such quick solutions support an idea that experienced programmers may avoid detailed semantic analysis of the problem statement and extract meanings based on established patterns [5]. Meaning mining is closely coupled with recognition of the problem variables. In many cases the input specification explicitly defines them. Let us consider a typical problem from http://ac.timus.ru archive:

Table 1. A typical trivial algorithmic problem

Title	1293. Eniya
Preamble	A story about reconstruction of a small ironclad space corvette “Eniya” is introduced.
Statement	N rectangular panels of $A \times B$ meters are processed by 1 nanogramm of thorium sulphide per square meter.
Input	Contains integers N ($1 \le N \le 100$), A ($1 \le A \le 100$), B ($1 \le B \le 100$).
Output	Weight of the needed thorium sulphide.

It is straightforward to parse the Input section using delimiting characters ‘(’, ‘)’, ‘≤’ and formally define all variables, including their types, names and ranges. This approach, however, has two major drawbacks. Firstly, not all problems specify the Input and / or Output sections in such explicit format. Secondly, and more importantly, the variable meanings and relationships between them remain unclear.

3. EXTRACTING VARIABLES

It has been claimed that repetitive observation of patterns and generalization drive learning of the grammar of a native language by kids [6]. Experiments show that frequent occurrence of a word sequence in linguistic environment determines its adoption and adaptation by children in a repetitive task [7]. Motivated by these observations, we propose an empirical algorithm that extracts the most meaningful tokens based solely on descriptive statistics and recognizes preliminary connections between them:

1. Tokenize the problem statement;
2. Exclude irrelevant grammatical units, such as prepositions, modal verbs, etc;
3. Count the frequencies of the remaining tokens;
4. Select the most frequent tokens or, assuming compact size of problem statements, those occurring at least three times. It is reasonable to expect the key meanings among them.
5. For each selected token collect all sentences it occurs in;
6. Group all tokens that occur exactly in the same sentences – they may constitute indices of some array;
7. Attach to each group of indices a token that occurs in the most of the group sentences – it defines the array name.

The outlined algorithm aims at recognition of arrays and logically connected concepts that enumerate the indices. Focusing on small and simple problems, we expect that at most one such array can be defined. If succeeded, the solution of the problem will be searched among array operations that satisfy the specified Input / Output format. Also, the equivalence relation between the concepts representing the indices will be saved for reuse.

4. TESTING RESULTS

To test the algorithm, we developed a web application working with problems from the same http://ac.timus.ru archive. Several cases are discussed below.

Table 2. The outline of Test Case 1.

Title	1409. Two Gangsters
Preamble	A story about Harry and Larry shooting at beer cans is introduced.
Statement	One can is shot by both, while others – by exactly one of them.
Input	The number of cans shot by Harry and by Larry respectively.
Output	The number of cans that were missed by Harry and by Larry respectively.

The problem statement is symmetric relative to tokens “Harry” and “Larry”, which, therefore, play identical roles. The algorithm generates the following statistics:

Table 3. The statistics of Test Case 1.

Token	Frequency	Sentences	Category
Cans	11	9	array
Larry	9	7	index
Harry	7	7	index

All three tokens appear in the same seven sentences with “cans” – twice more elsewhere. Therefore, they are combined in an array cans[2], with cans[0] representing “cans of Harry”, and cans[1] – “cans of Larry”. The order is decided by the Input format.

Table 4. The outline of Test Case 2.

Title	1573. Alchemy
Preamble	A story about potions and reagents of red, blue and yellow colors is introduced.
Statement	The quantities of B blue, R red and Y yellow reagents, and the potion recipe are given.
Input	Contains integers B, R, and Y; $1 \leq B, R, Y \leq 100$. Then colors of the required reagents (“Blue”, “Red”, or “Yellow”) come. Each word occurs at most once.
Output	The number of possible ways to choose a set of reagents from.

The problem statement is again symmetric, this time relative to tokens “red”, “blue” and “yellow”. An important step here is to regularize the past forms of the verbs, to avoid such occurrences as “required” and “considered”. The algorithm generates the following statistics:

Table 5. The statistics of Test Case 2.

Token	Frequency	Sentences	Category
reagents	10	7	array
potions	7	7	array
red	7	7	index
blue	7	7	index
yellow	7	7	index

The last three tokens appear in the same seven sentences, where “reagents” occurs three times and “potions” – only twice. Therefore, the indices are combined in a 3-element array reagents[3], with reagents[0] representing “blue reagents”, reagents[1] – “red reagents”, and reagents[2] – “yellow reagents”. Again, the order is decided by the Input format.

5. CONCLUSIONS

We studied meaning extraction from statements of easy algorithmic problems. A statistical algorithm is developed and tested for certain type of such problems that allow enumeration-based solution. We looked at problems from the http://acm.timus.ru archive. Unfortunately, only 4 out of more than 150 easy problems exhibit such nature. So, more sources are required for deeper testing.

REFERENCES

[1] J. Tang, et al. "Information extraction: Methodologies and applications" *Emerging Technologies of Text Mining: Techniques and Applications,* 2007.

[2] E. Westerhout. "Definition extraction using linguistic and structural features" *Proc. of the 1st Workshop on Definition Extraction,* Association for Computational Linguistics, pp. 61-67, 2009.

[3] I. S. Bajwa. "Context based meaning extraction by means of markov logic" *International Journal of Computer Theory and Engineering*, 2.1: 1793-8201, pp. 35-38, 2010.

[4] T. Vasiga, et al. "Structure, scoring and purpose of computing competitions" *Informatics in Education - An International Journal*, 5_1, pp. 15-36, 2006.

[5] B. Delibasic, K. Kirchner, J. Ruhland. "A pattern based data mining approach" *Data Analysis, Machine Learning and Applications*, Springer Berlin Heidelberg, pp. 327-334, 2008.

[6] M. Tomasello. "*Constructing a language: A usage-based theory of language acquisition*", Cambridge, MA: Harvard University Press, 2003.

[7] C. Bannard, D. Matthews. "Stored Word Sequences in Language Learning: The Effect of Familiarity on Children's Repetition of Four-Word Combinations." *Psychological science*, 19.3, pp. 241-248, 2008.